



ACTIVATE BUSINESS WITH THE POWER OF I.T.™



# DB2® 9 for z/OS What's In It For You?

Bill Arledge – Consulting DB2 Product Manager

9/5/2008

# Agenda



- › Introduction
- › DB2 9 Quick Overview
- › Infrastructure Changes
- › PureXML™

# An Introduction



## › Our Goal

- More DB2 9 Knowledge means more timely and effective migrations
- Implementation Considerations are often overlooked in the technical details

## › Caveats

- One hour is hardly enough time to cover DB2 9 in detail so we'll be covering infrastructure highlights and a little bit on XML
- DB2 9 is fluid with more being learned and new features being delivered

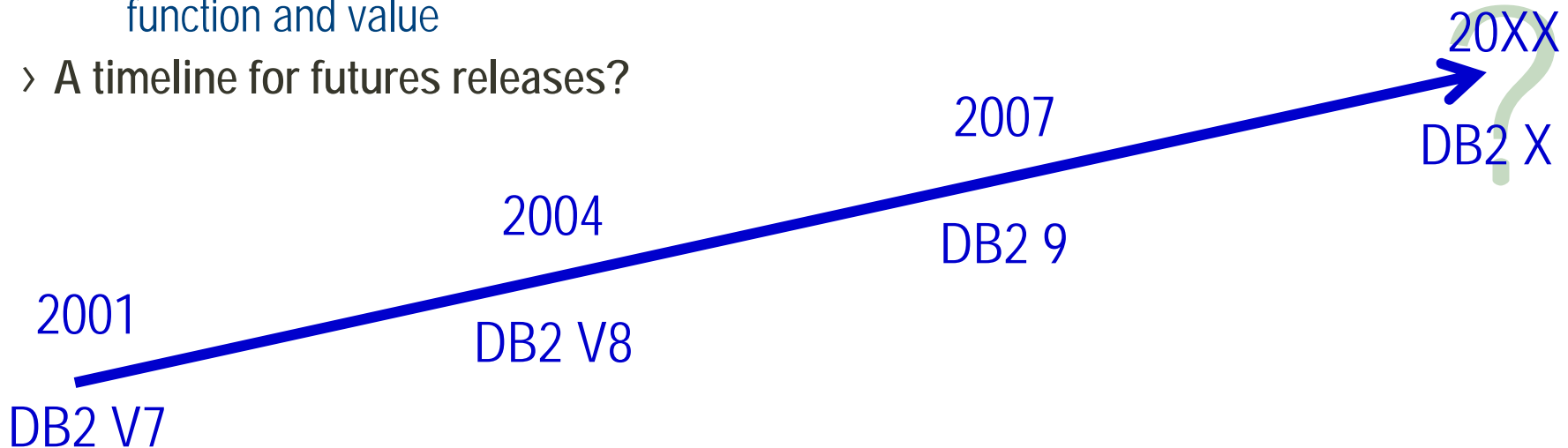
## › BMC's DB2 9 History

- Initial evaluation of DB2 began in early 2005
  - On Paper
- ESP Support in 2006
  - Support for multiple customers and multiple products
- DB2 9 support in August 2007

# Current DB2 Status



- › DB2 V8 in the field four years
  - Many customers completing migration to DB2 V8 this year
  - Major migration effort
- › DB2 9 available 13 months
  - Migrations ramping up in 2009
  - Less difficult than V8
  - A major release with significant new function and value
- › A timeline for futures releases?



# DB2 9 – A Quick Overview



## › Overall Value – The Marketing Perspective

- *Decreased cost of ownership*
- *Increased consistency with other DB2 platforms*
- *Increased availability through database definition on demand*
- *Better performance*
- *Improved regulatory compliance*
- *Increased synergy with System z™*

## › What Delivers the Value

- *Additional 64-bit exploitation*
- *PureXML™*
- *New SQL functionality*
- *More performance options*
- *Better availability*
- *New security features*

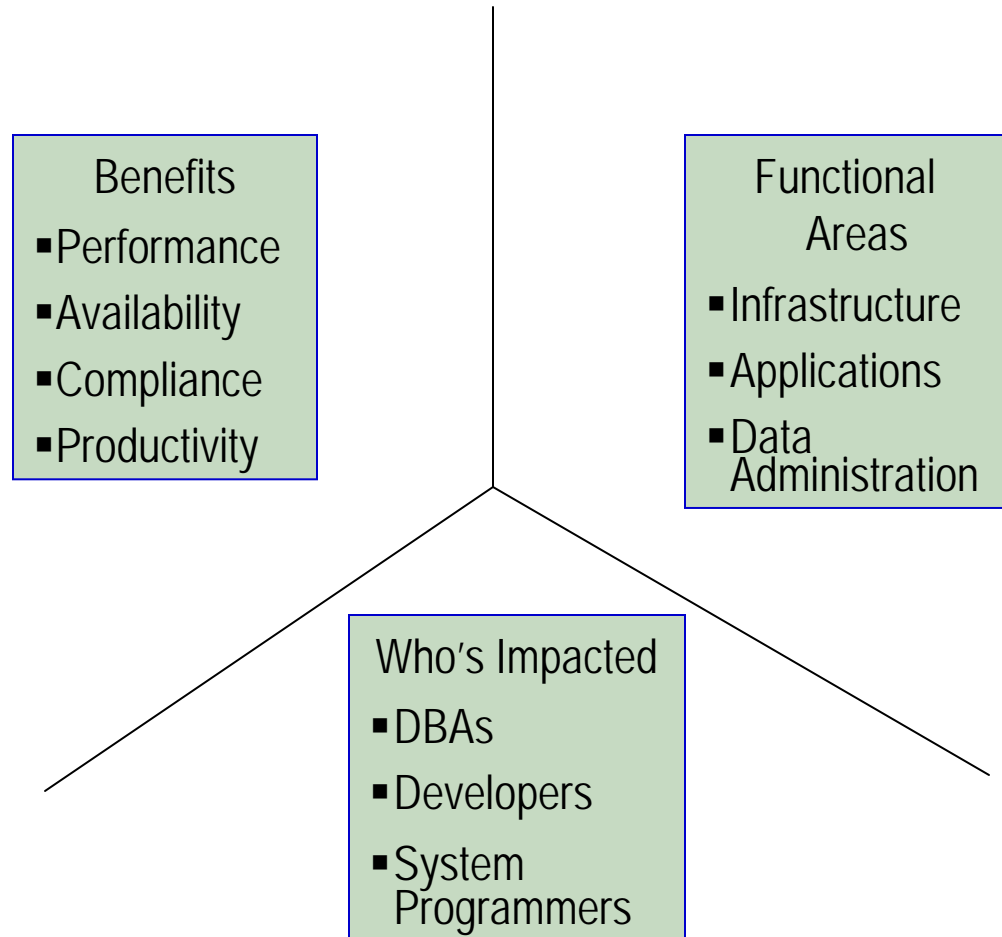
# DB2 9 Details

## How do you Look at it?



### › Different Angles

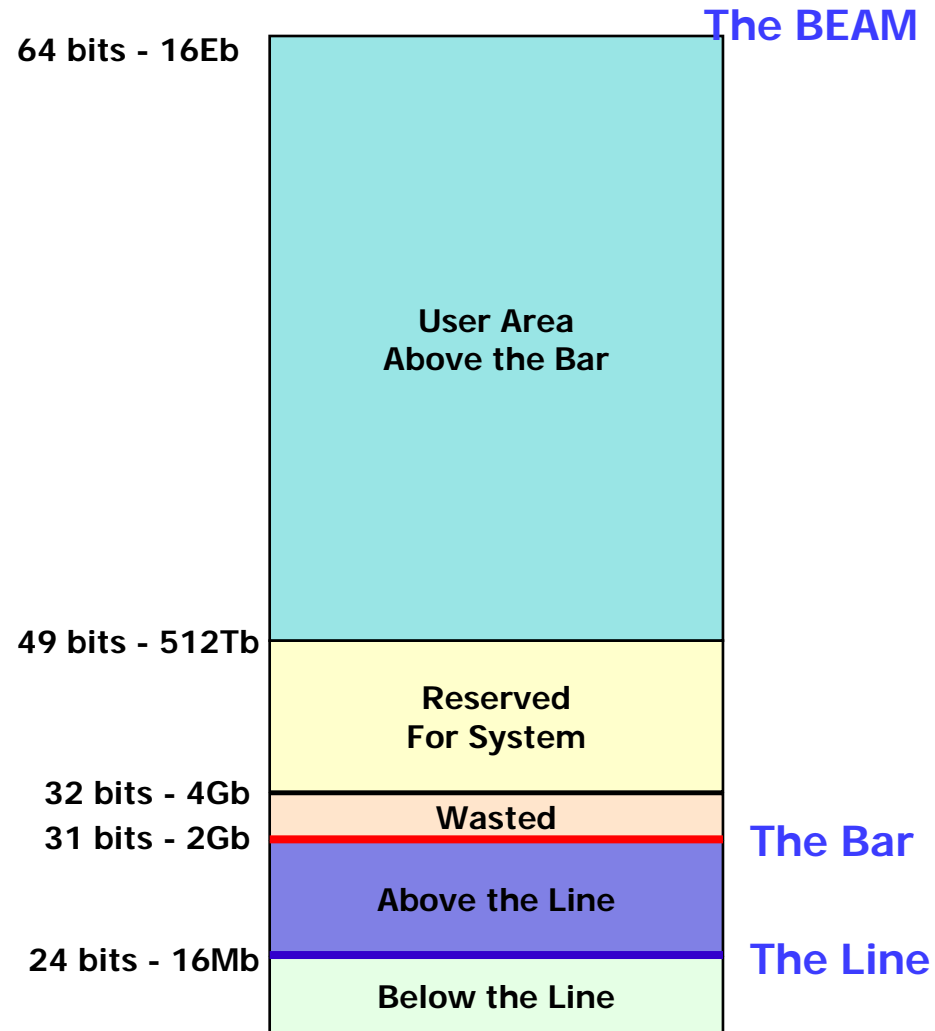
- DB2 9 from different perspectives
- Presentation organized by functional area with comments on benefits and implementation considerations



# Infrastructure Changes Extended 64-bit Support



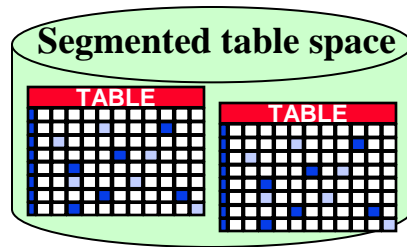
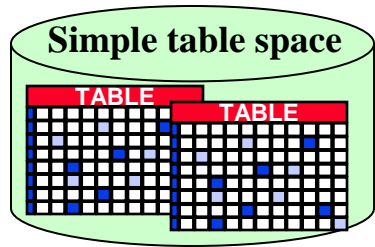
- › **DIST Address Space now utilizing above the bar storage**
  - Provides for VSCR in DBM1 and DIST address spaces.
  - New shared memory object reduces CPU
- › **EDMPOOL changes**
  - Portions of runtime Components moved above the bar
    - Plan and package skeletons above the bar
    - Bound/Prepared DML Statements
      - Statement Text
      - SQLDA DESCRIBE output
      - Portion of native SQL PL package
    - Portions of static SQL sections (CT/PT) are moved as well
    - Requires rebind to realize full benefit
  - Further reduces VSC in the DBM1 address space
    - Reduces below-the-bar storage by 10% - 15% on average
- › **Provided in CM (Compatibility Mode)**



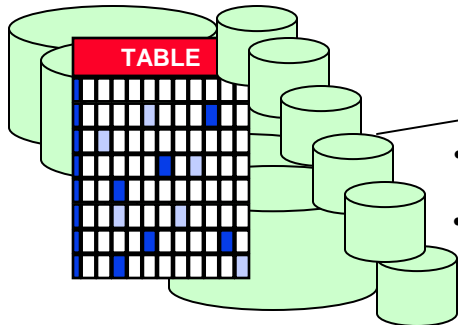
# DB2 9 Infrastructure Table Space Changes



## DB2 V8



## Partitioned table space



- Index-controlled partitioning
- Table-controlled partitioning
- Strategic

## DB2 9

- Deprecated (means existing simple TS continue to be supported but can no longer be created)
- Becomes the "default" table space type
- Traditional Partitioned table spaces Remain
- New UTS (Universal table space)
  - Partition-by-Growth (PBG)
    - ❖ More partitions as data grows
  - Partition-by-Range (PBR)
    - ❖ Similar to existing partitioned tablespaces
  - Combines benefits of segmented and partitioned organizations

# DB2 9 Infrastructure Universal Table Space – Partition-by-Growth



› New partitions automatically created as allocated partitions fill-up

```
CREATE TABLESPACE WDATB1TS IN DSNDB04
```

```
.....
```

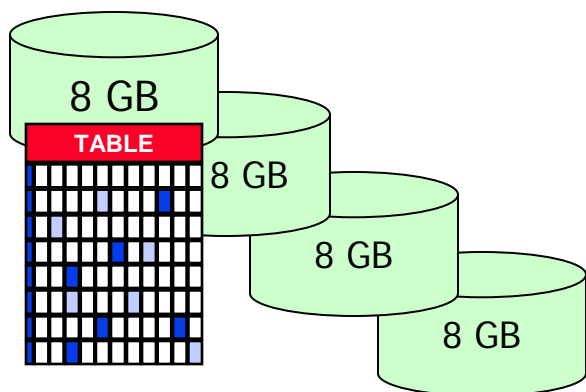
```
DSSIZE 8G MAXPARTITIONS 64
```

• table space CREATE for PBG

DBNAME	TSNAME	TYPE	SEGSIZE	DSSIZE	MAXPARTITIONS
DSNDB04	WDATB1TS	G	32	8388608	64

'G' Indicates partition-by-growth

• What's in SYSIBM.SYSTABLESPACE



Partition-by-Growth table space

```
PARTITION BY SIZE EVERY 8 G
```

• Controlling DSSIZE for implicit TS

```
ALTER TABLESPACE DSNDB04.WDATB1TS  
MAXPARTITIONS 128;
```

• Allowing for more partitions  
• Can't lower the number

# DB2 9 Infrastructure

## Universal Table Space – Range-Partitioned



- › Functions like traditional partitioned table space in DB2 V8
  - Uses segmented space management
    - Indicated by a SEGSIZE > 0

```
CREATE TABLESPACE CRIN2RTS IN CRRMGRDB
DSSIZE 4G SEGSIZE 32 NUMPARTS 3
```

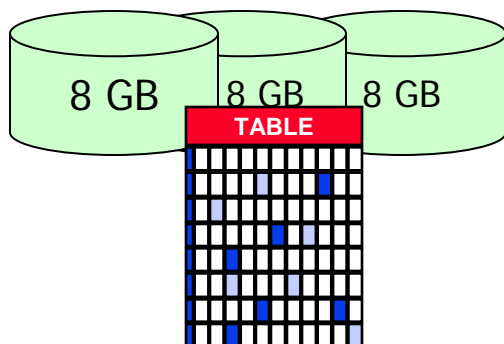
• table space CREATE for PBR

DBNAME	TSNAME	TYPE	PARTITIONS	SEGSIZE	DSSIZE
CRRMGRDB	CRIN2RTS	R	3	32	4194304

• What's in SYSIBM.SYSTABLESPACE

'R' Indicates range-partitioned

Range-partitioned table space



```
CREATE TABLE MKTCWR.RM_IND_2
(PRIMARY_KEY_A CHAR(2) NOT NULL
,PRIMARY_KEY_B CHAR(10) NOT NULL
.....
IN CRRMGRDB.CRIN2RTS
PARTITION BY RANGE
((PRIMARY_KEY_A ASC,PRIMARY_KEY_B ASC)
(PARTITION 1
ENDING AT ('01')
,PARTITION 2
ENDING AT ('05')
,PARTITION 3
ENDING AT ('10'))))
```

# DB2 9 Infrastructure

## Universal Table Space Considerations



### › Benefits

- Segmented format benefits include:
  - Better space management for varying-length rows
  - Improved mass delete operations
- Improves availability and simplifies management process for IT staff

### › Implementation Considerations

- Single table objects
- Drop/Create required for implementation
- UTS required for other DB2 9 features
  - Clones, XML auxiliary TS
- UTS Partition-by-growth is default for implicit table spaces
  - PARTITION BY SIZE syntax controls DSSIZE (defaults to 4GB)
- UTS Partition-by-growth TS MAXPARTITIONS can be increased via ALTER
  - No ability to reduce number
- Reorganization may reduce number of required partitions but defined partitions will remain in the catalog and minimally formatted

# DB2 9 Infrastructure

## NOT LOGGED Table Spaces



### › Allows you to turn off logging for specific table space objects

- CREATE or ALTER syntax
- Propagates to dependent objects
  - Indexes, auxiliary objects

```
CREATE TABLESPACE CRINDDTS IN CRBMCDDB
LOGGED / NOT LOGGED
*****
```

### › Candidates include summary tables, MQTs, etc.

### › What's the benefit?

- Performance seems an obvious benefit; but,
  - Application logging is usually not a performance issue
    - Parallel insert processes are one possibility for avoiding bottlenecks on the log
    - Performance measurements show small improvements in class 2 CPU and elapsed times
  - Overall reduction in amount of log data could improve recovery processes

### › Implementation Considerations

- Commit frequency is very important
  - Restart processing is still required even without logging
- Abort or Rollback requires recovery to a previous point of consistency
  - All updates are lost (yours and theirs)

# DB2 9 Infrastructure

## Reordered Row Format



### › A Little History on variable length columns

- Additional CPU overhead required to cross the variable length column to get to the desired column
- Physical design recommendation to place variable length columns at the end of the row
  - Similar recommendation to place frequently update fixed-length columns at end
- Controlling placement of columns is more challenging in today's environment

### › DB2 9 changes the way variable length columns are stored and managed

- All fixed length columns placed at front of row
- 2 byte offset values for each VARCHAR column follow
- Variable column data at the end of the row in storage

# DB2 9 Infrastructure

## Reordered Row Format



- Create Table NAMEADDRESS

```
(Firstname      Varchar(64) Not Null,
  Lastname      Varchar(64) Not Null,
  Address1      Char      (20) Not Null,
  Address2      Char      (20) Not Null)
```

### Old format on Disk

Length	Firstname VC	Length	Lastname VC	Address1 CHAR(20)	Address2 CHAR(20)
x'0007'	Belinda	x'0006'	Nelson	Morado Circle bbbbbbbb	Austin bbbbbbbbbbbbbbbb

### New Format on Disk:

Address1 Char (20)	Address2 Char(20)	Offset	Offset	Firstname VC	Lastname VC
Morado Circle bbbbbbbb	Austin bbbbbbbbbbbbbbbb	x'002C' (44bytes)	x'0033' (51bytes)	Belinda	Nelson

# DB2 9 Infrastructure

## Reordered Row Format



### ›Implementation Considerations

- Standard with DB2 9 New Function Mode – feature cannot be turned off
- All table spaces created on DB2 9 NFM will be in RRF; except:
  - Catalog Tables, LOB table spaces and tables with EDITPROCs or VALIDPROCs
    - Good documented approaches for handling migrations with these procedures
- Migrated objects will have Basic Row Format (BRF) until a reorganization or Load Replace occurs
  - REORG automatically regenerates compression dictionary even with KEEPDICTIONARY
- SQL results columns returned in order of columns on the CREATE statement

### ›Benefits

- CPU cost reduction for processing variable length rows
- Logging should be roughly the same
  - Potential increase in logging if table design tweaked to reduce logging
    - Volatile fixed-length columns at end of table for instance

# DB2 9 Infrastructure Cloned Tables



- › Provides for fast replacement of application data
  - Online Load Replace

- › Variations on the ALTER TABLE SQL statement are used to add or drop clones

```
ALTER TABLE MKTCWR.RM_IND_2
ADD CLONE MKTCWR.CLONE_RM_IND_2;
```

- › Datasets created

```
DSNDIA.DSNDBD.CRRMGRDB.CRIN2RTS.J0001.A001
DSNDIA.DSNDBD.CRRMGRDB.CRIN2RTS.J0002.A001
DSNDIA.DSNDBD.CRRMGRDB.CRIN2RX1.J0001.A001
DSNDIA.DSNDBD.CRRMGRDB.CRIN2RX1.J0002.A001
```

- › Catalog Information
  - SYSTABLESPACE

NAME	CLONE
CRIN2RTS	Y
IAUDXLRP	Y

- SYSTABLES

Table Name	Database	Tblspace	ColsPK	Type
MKTCWR.CLONE_RM_IND_2	CRRMGRDB	CRIN2RTS	8 2	C
MKTCWR.RM_IND_2	CRRMGRDB	CRIN2RTS	8 2	T

-DISPLAY command output

NAME	TYPE	PART	STATUS
CRIN2RTS	TSB1	0001	RW
-THRU		0003	
CRIN2RTS	TSC2	0001	RW
-THRU		0003	
CRIN2RX1	IXB1	0001	RW
-THRU		0003	
CRIN2RX1	IXC2	0001	RW
-THRU		0003	

# DB2 9 Infrastructure

## More on Clones



### › Implementation Considerations

–Contents of the base table and associated clone can be switched

- EXCHANGE DATA BETWEEN TABLE tname1 AND tname2

Table Name	Database	Tblspace	Cols	PK	Type
MKTCWR.CLONE_RM_IND_2	CRRMGRDB	CRIN2RTS	8	2	C
MKTCWR.RM_IND_2	CRRMGRDB	CRIN2RTS	8	2	T

–SQL INSERTS and Load Replace can populate the clone

–Most utilities can operate on the clone using new CLONE keyword

- Statistics are not maintained on the clone
  - If you expect differences in statistics between with the clone then you'll need to collect statistics after the EXCHANGE DATA operation

–Base table can be recovered from an image copy taken on the clone

### › Benefits

–Improved availability of the base table during a Load Replace type operations

# DB2 9 Infrastructure

## Compressed Indexes



› New capability for compressing indexes

› Implementations Considerations

– New syntax on CREATE or ALTER INDEX

- COMPRESS YES/NO
- ALTER will place the index in REBUILD PENDING

```
ALTER INDEX RNDWDA.CRIN2RX1  
COMPRESS YES;■
```

– No compression dictionary used

– Index must specify a 8, 16, or 32K buffer pool

- Always compresses down to a 4K page on disk

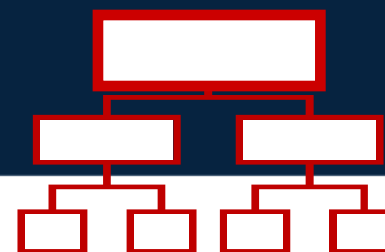
– Index leaf page is only index page compressed on disk

- Page is always decompressed in the buffer pool
- Compression and decompression occur during disk I/O

– Index changes are logged uncompressed

– DSN1PRNT supports printing of index pages in compressed or uncompressed format

# DB2 9 Infrastructure Compressed Indexes



## › Benefits

- Major benefit is disk savings
  - Important where more and more indexes are being built
  - Many times required index storage surpasses table space storage
  - Averages compression is around 50% (as per Redpaper)
    - Use DSN1COMP to determine likely savings and identify best buffer pool size
- I/O should go down
- CPU cost will go up for both the application and the DBM1 address space

# DB2 9 Infrastructure

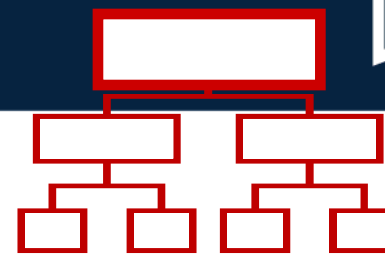
## Index On Expression



- › Index definitions can now include expressions
- › An “Extended Index”
  - Info stored in `SYSKEYTARGETS` with and indicator in `SYSINDEXES`
    - `KEYTARGET_COUNT`>0
- › Implementation Considerations
  - Restrictions:
    - Resulting data type cannot be LOB or XML
    - Expression must contain at least one column from the table
      - Column cannot be a user defined datatype
  - Expression can be a maximum of 4000 bytes when converted to UTF-8
  - 64 expressions in a single index definition
  - `ALTER INDEX` with `REGENERATE` allows altering an expression and forces index into `REBUILD PENDING`

# DB2 9 Infrastructure

## More Index On Expression



> CREATE INDEX TAXCONTRIB ON TABLEA  
(EMPNAME,(SALARY \* (TAXRATE/100)));

### SYSINDEXES:

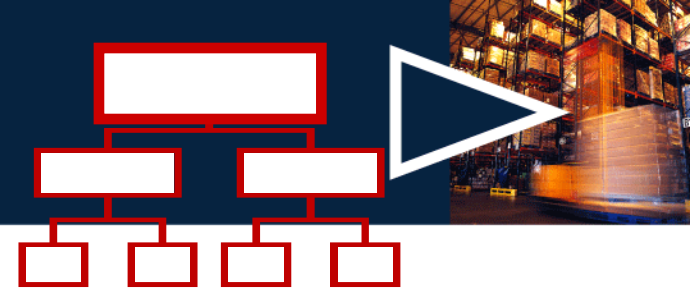
Colname	Colno	Ordering
EMPNAME	1	blank
SALARY	2	blank
TAXRATE	3	blank

### SYSKEYTARGETS:

Keyseq	Colno	Datatypeid	Length	Derivedfrom
1	1	448(vchar)	128	EMPNAME
2	0			SALARY*(TAXRATE/100)

# DB2 9 Infrastructure

## Index On Expression

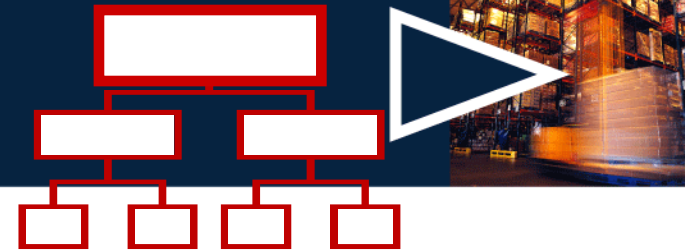


### › Benefits

- Major performance improvements are possible
  - Predicates with expressions can now be Stage 1
  - Significant CPU and GETPAGE reductions for queries
- Some increase in overhead for updating SQL statements required for evaluating index expressions
  - INSERT and UPDATE
- Some increase in overhead for processes requiring index maintenance
  - LOAD, REBUILD INDEX, REORG TABLESPACE
  - Overhead not considered significant

# DB2 9 Infrastructure

## Relief For Sequential Insert



- › Improves DB2's management of index page splits
  - Prior to DB2 9 index page splits followed these rules
    - Page splits at the end of the index used a 100/0 algorithm
    - Page splits in the middle of the index used a 50/50 algorithm
      - For hot spots in the data with intensive INSERT processing many page splits would occur and leave many index pages half full
- › **Implementation Considerations**
  - DB2 9 detects a pattern of sequential inserts in the middle of an index and changes from 50/50 to a 90/10 approach for managing page splits
    - Known as an asymmetric page split
    - Reduces number of half-empty pages that will never be inserted into
  - Automatic function available in Compatibility Mode
- › **Benefits**
  - More efficient index space management
  - I/O reduction

# DB2 9 Infrastructure

## Larger Index Page Sizes



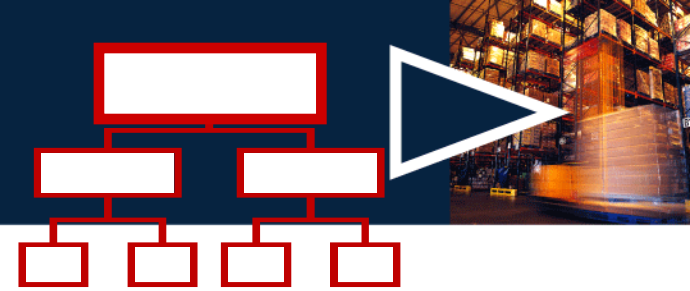
- › DB2 9 support additional index page sizes
  - 8K, 16K, and 32K
- › Implementation Considerations
  - CREATE and ALTER INDEX syntax
  - CREATE and ALTER DATABASE syntax
  - Buffer pool identifies page size
  - Required for compressed indexes
- › Benefits
  - Reduces the need for index page splits
    - More keys on the index page
  - Improves compression

```
ALTER INDEX RNDWDA.CR_INDIV_IX1  
BUFFERPOOL BP8K1;
```

```
CREATE DATABASE CRBMCDDDB  
BUFFERPOOL BP0  
INDEXBP BP16K0  
STOGROUP MKTCWRSG  
CCSID EBCDIC;
```

# DB2 9 Infrastructure

## Random Indexes



### › Hot spots in indexes can cause contention

- Data sharing environments where concurrent INSERTs against the same index pages create contention

### › Implementation Considerations

- CREATE and ALTER INDEX syntax
- Limitations

```
CREATE INDEX RNDWDA.TSTINDX2 ON RNDWDA.RM_IND_2
(CUST_LAST_NAME ASC,
CUST_FIRST_NAME ASC,
CURR_ORDER_DATE RANDOM);■
```

- Not valid for a varying-length index key column in a NOT PADDED index
- Indexes with RANDOM columns can't be used for RANGE SCANS
- Can be used for non-matching index scans and index-only access

### › Benefits

- Reduced contention in data sharing environments
- Can increase getpages, lock requests, and I/Os
- Potential improvement or degradation

# DB2 9 Infrastructure

## Automatic Creation of Objects (Autonomic DDL)



- › Impractical for non-DBAs to create additional objects required to support a new user-defined table
  - Prior to DB2 9 table spaces could be implicitly created
- › With DB2 9 the following additional objects are implicitly created
  - Database
    - DSNDB04 is no longer the default database
    - Implicit database is created from defined range of database names
      - DSN00001 – DSN10000 as of PK62178
  - Partition-by-growth tablespace
    - Partition-by-Range will fail without recent maintenance
  - Enforcing primary key
  - Enforcing unique key
  - ROWID index (for GENERATED BY DEFAULT)
  - LOB table space, auxiliary table and auxiliary index

# DB2 9 Infrastructure

## Automatic Object Creation



- › CREATE TABLE with no DATABASE or TABLESPACE specification

```
CREATE TABLE RNDWDA.CR_ORDERS
(PRIMARY_KEY_A CHAR(2) NOT NULL
.....)
```

### Implicit database characteristics

- STOGROUP SYSDEFLT
- Index and tablespace buffer pools based on TBSBP\* and IDXBP\* DSNZPARAM values

- › Objects automatically created

### – Database

```
Database Owner  Stogroup Buf
---1---v---2---v---3---v---
DSN00066 SYSIBM  SYSDEFLT BPO
```

### – Table space

```
TSNAME  DBNAME  SEGSIZE  MAXPARTITIONS
-----+-----+-----+-----
CRRORDER DSN00066      4          256
```

### – Table

```
TBLNAME  CREATOR  TYPE  DBNAME  TSNAME
-----+-----+-----+-----+-----
CR_ORDER RNDWDA    T     DSN00066 CRRORDER
```

### Implicit table space characteristics

- SEGSIZE 4
- UTS partition-by-growth
  - DSSIZE 4G unless overridden
    - PARTITION BY SIZE EVERY n G
- UTS range-partitioned
  - If partition keys provided in CREATE TABLE
  - Recent maintenance makes this work
    - APAR PK51752 (PTF UK34665)
- DSNZPARAMs control tablespace defaults
  - IMPDSDEF – DEFINE YES/NO
  - IMPTSCMP – COMPRESS YES/NO

# DB2 9 Infrastructure

## Automatic Object Creation



### › Benefits

- Big Productivity Boost in Ad hoc Environments
  - Non DBA end-users can create the own structures from top to bottom, decreasing IT administration involvement
- Improved portability to other platforms

### › Implementation Considerations

- Implicit databases use a sequential numbering scheme
  - *No explicit databases of this form can be created*
  - *DSN prefix can be used for other databases*
    - *Sequential numbers > 10000 can not be used*
- Dropped implicit databases will not be reused until the number turns over



- › APPEND tells DB2 to disregard the clustering sequence during INSERT and online load processing
  - Inserted rows are place at the end
    - For range-partitioned rows are placed at the end of the partition for that key value
    - For partition-by-growth rows are placed at the end of any partition where space is available
- › APPEND YES on CREATE or ALTER statements
  - New column in SYSTABLES
- › Clustering sequence will be impacted but a reorganization can be used to reestablish
- › Benefits online load utility execution and batch processes with heavy INSERT activity

```
ALTER TABLE  
RNDWDA.CR_ORDERS  
APPEND YES
```

# DB2 9 Infrastructure

## Rename Index



### › Allows for renaming of indexes

- New option for the extended RENAME statement

```
RENAME INDEX RNDWDA.TSTINDX2  
TO TSTINDX5
```

- Old index name disappears from the catalog during the process

### › Implementation Considerations

- All bound packages and plans refer to the OBID so there isn't any issue
- Index names appear in a few places
  - Plan table entries reference specific indexes
  - Index names are stored in the dynamic statement cache
  - Any explicit SQL references to the index will need to be modified after the change

### › Benefit

- Avoids dropping and rebuilding an index to change the name

### › A Side Comment

- Online Schema Evolution has evolved to Database Definition on Demand

# DB2 9 Infrastructure

## Rename Column



### › Allows for renaming of table columns without a drop/create process

- New option for the ALTER TABLE statement

```
ALTER TABLE RNDWDA.RM_IND_2  
RENAME COLUMN CUST_FIRST_NAME  
TO CUST_FRST_NM
```

- Existing objects associate with the column will function the same

### › Implementation Considerations

- All plans/packages that reference the table will be invalidated
- Applications that reference this column will need work
- Some limitations for this function (too many to discuss here)
  - For example: ALTER TABLE RENAME COLUMN is not allowed when the source column is referenced in a view

### › Benefit

- Avoids dropping and rebuilding the table in order to change the name

# DB2 9 Infrastructure

## STOGROUP SMS Constructs



- › DB2 9 introduces three new SMS attributes for STOGROUPs
  - DATACLAS, MGMTCLAS, STORCLAS
- › Previously, for STOGROUP defined objects the only way provide these attributes was via complex ACS routines
  - Based on dataset names
- › SMS attributes can be assigned during CREATE or ALTER processing

– ALTER Syntax

```
CREATE STOGROUP MKTCWRSG
VOLUMES (SYSBXX)
VCAT DSN DIA
```

```
DATACLAS DCLASS01
MGMTCLAS MCLASS02
STORCLAS SCLASS03
```

New attributes (columns) in  
SYSIBM.SYSSTOGROUP

- › Benefits – Increases the flexibility for handling dataset allocation and reduces manual effort required

# DB2 9 Infrastructure Enhancements

## REOPT(AUTO)



- › New bind option that allows reoptimization of dynamic statements when DB2 determines that filter factors for a specific predicate are significantly different
  - Prepared dynamic SQL statements end up in the dynamic statement cache (depending on bind options) and are reused until they leave the cache
  - Multiple bind options control the behavior of dynamic statements assuming caching is turned on
    - ALWAYS – statements not eligible for caching
    - ONCE – SQL is reoptimized with the first set of input variables; dynamic SQL is placed in the dynamic statement cache
    - NONE – Says use the access path determined during the bind or prepare; statement is cached
- › New REOPT(AUTO) let's DB2 determine if another access path with better performance characteristics is possible
- › Benefits and Implementation Considerations
  - Could dramatically improve performance
  - REOPTTEXT DSNZPARM must be set to YES (default is NO)
  - Most potential where query performance varies widely with the parameter marker values

# DB2 9 Infrastructure Management

## PLANMGMT – Plan Stability



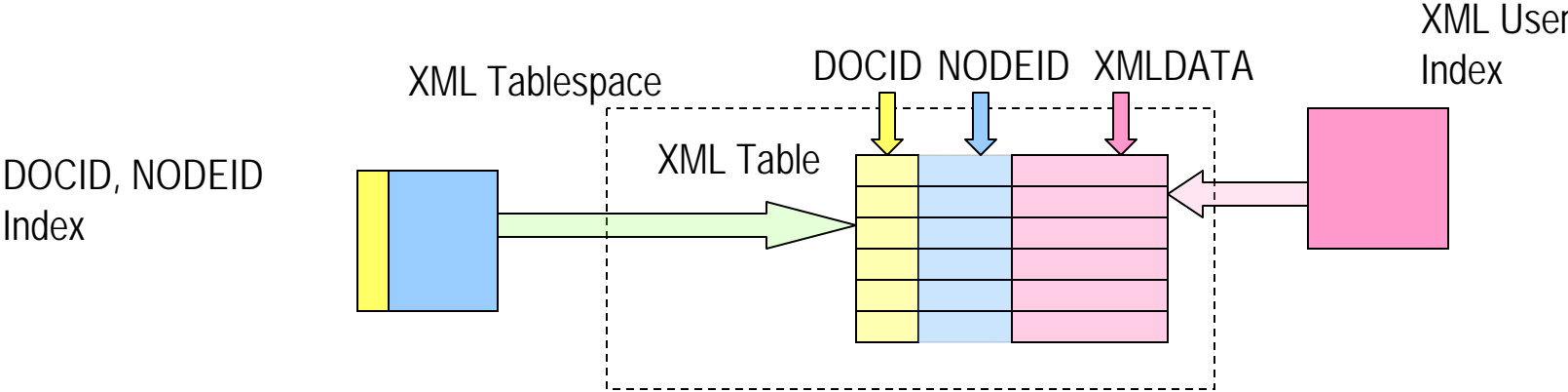
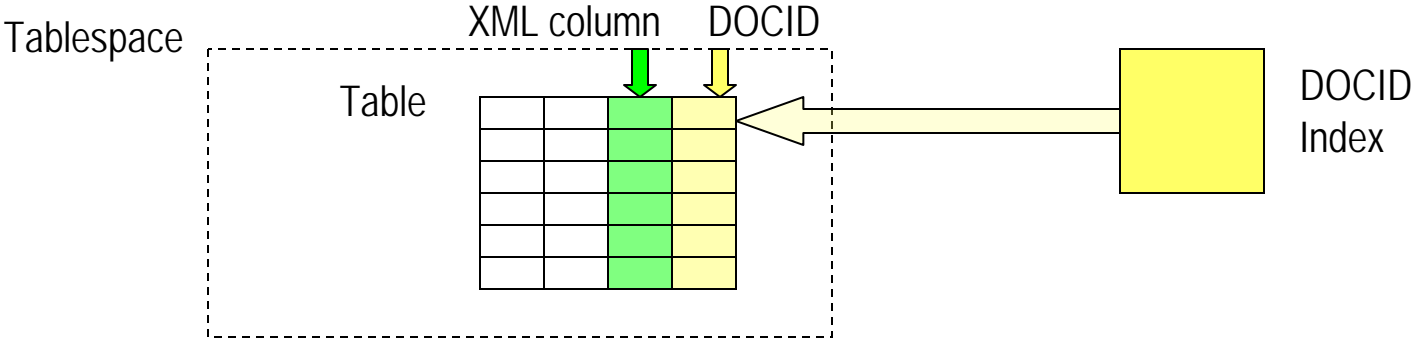
- › New option for maintaining multiple versions of bound packages along with the ability to switch to a previous copy
  - Multiple bind options control the behavior of dynamic statements assuming caching is turned on
  - Delivered after DB2 9 availability (APAR PK52523 (PTF UK31993))
- › How's it work
  - PLANMGMT DSNZPARM turns it on for the subsystem
  - New bind options for REBIND PACKAGE and REBIND TRIGGER PACKAGE
    - PLANMGMT option determines number of package copies to keep
      - OFF - keeps only the active copy of the bound package
      - BASIC - keeps the active copy and one previous copy if one exists
      - EXTENDED - keeps active copy, one previous copy, and the original page copy
    - SWITCH bind option allows you to fallback to the PREVIOUS or ORIGINAL copy
      - Which now becomes the ACTIVE copy
- › Benefits and Implementation Considerations
  - Rebind can occasionally lead to degraded performance; this means you can fallback
  - Multiple package copies will double or triple the size of SPT01



- › IBM refers to it as PureXML
- › Allows storage of XML in DB2 tables
  - New datatype in columns is XML
  - Can be accessed with SQL or XQUERY
- › DB2 hides complexity from user
  - Hierarchical DBMS on top of DB2
- › All the reliability, security, recoverability of the mainframe, with the flexibility and accessibility of XML



# DB2 9 Infrastructure XML Spaces



# DB2 9 Infrastructure

## XML - Implementation Considerations



### › Performance

- Data design is still necessary!
- “Extract” static metadata to relational columns
- Don’t use XMLQUERY without an XML index
- UPDATE is really delete/insert of XML column
- LOAD uses INSERT like process



# DB2 9 Infrastructure XML Implementation Considerations



## › Maintenance

- Finding names of implicitly created objects is non-trivial
- Use ALL option in LISTDEF
- LOAD must use files for XML columns larger than 32K

## › Migration (DW, Test creation)

- Element and attribute names are not stored with data
- SYSIBM.SYSXMLSTRINGS matches the id actually stored with the name
- DSN1COPY is not an option

# More DB2 9 in IBM Publications



- › DB2 9 Technical Overview SG24-7330
- › DB2 9 Performance Topics SG24-7473
- › DB2 9 Stored Procedures SG24-7604
- › Index Compression with DB2 9 for z/OS paper
- › z/OS XML System Services Synergy Update
- › SQL Reference for Cross-Platform Development
- › LOBs with DB2 for z/OS SG24-7270
- › Powering SOA with IBM Data Servers SG24-7259
- › Enhancing SAP – DB2 9 SG24-7239
- › Best Practices SAP BI – DB2 9 SG24-6489-01
- › Data Sharing in a Nutshell SG24-7322
- › Securing DB2 & MLS z/OS SG24-6480-01

# Summary



- › Another major DB2 release from IBM
  - Migration not a challenging as DB2 V7 to DB2 V8
- › Still light on the uptake
  - BMC numbers indicate less than 25% have migrated as we speak
  - Expect significant increases later this year and in 2009
- › Extensive education on DB2 9; from:
  - IBM
  - DB2 Customers
  - DB2 Vendors
- › BMC has a major commitment to DB2 and fully supports DB2 9 features
  - As well as the DB2 community worldwide